Amendments to the Claims

This listing of claims will replace all prior versions, and listings of claims in the application.

Listing of Claims

1. (Currently Amended) A method for storing a digital value to memory in a pipelined instruction processor, wherein the digital value is read from memory in response to a conditional jump instruction to determine if the condition of the conditional jump instruction is satisfied, the method comprising:

generating at least one status bit based on the digital value to be stored, the at least one status bit relating to a particular condition of a conditional jump instruction and specifying ~~indicating~~ if the particular ~~a predetermined~~ condition of [[a]] the conditional jump instruction is satisfied or not; and

storing the digital value and the at least one status bit to memory.

2. (Currently Amended) The method recited in claim 1, wherein the conditional jump instruction reads the digital value and the at least one status bit from memory to determine if the condition of the conditional jump instruction is satisfied without having to submit the condition of the conditional jump instruction to an arithmetic logic stage of the pipelines instruction processor.

3. (Previously Presented) The method recited in claim 1, wherein the at least one status bit is read from memory at the same time as the digital value.

4. (Previously Presented) The method recited in claim 1, wherein the memory has one or more addressable locations, and the at least one status bit is stored at the same addressable location as the corresponding digital value.

5. (Previously Presented) The method recited in claim 1, wherein the at least one status bit is set high if the digital value is zero.

6. (Previously Presented) The method recited in claim 1, wherein the at least one status bit is set high if the digital value is a positive value.

7. (Previously Presented) The method recited in claim 1, wherein the at least one status bit is set high if the digital value is negative.

8. (Previously Presented) The method recited in claim 1, wherein the at least one status bit is set high if the digital value is a non zero value.

9. (Previously Presented) The method recited in claim 1, wherein the at least one status bit is set high based on the value of the least significant bit of the digital value.

10. (Currently Amended) In a pipelined instruction processor that executes instructions including conditional jump instructions, one or more of the conditional jump

instructions reading a digital value from memory to determine if the condition of the conditional jump instruction is satisfied, the improvement comprising:

a status bit generator for generating at least one status bit based on a digital value, the at least one status bit relating to a particular condition of a conditional jump instruction and specifying ~~indicating~~ if ~~a predetermined~~ the particular condition of [[a]] the conditional jump instruction is satisfied or not; and

storing means for storing the digital value and the at least one status bit to the memory.

11. (Currently Amended) The pipelined instruction processor recited in claim 10, wherein a selected conditional jump instruction reads the digital value and the at least one status bit from memory to determine if the condition of the conditional jump instruction is satisfied without having to submit the condition of the conditional jump instruction to an arithmetic logic stage of the pipelines instruction processor.

12. (Previously Presented) The pipelined instruction processor recited in claim 10, wherein the at least one status bit is read from the memory at the same time as the digital value is read.

13. (Previously Presented) The pipelined instruction processor recited in claim 10, wherein the memory has one or more addressable locations, and the at least

one status bit is stored at the same addressable location as the corresponding digital value.

14. (Previously Presented) The pipelined instruction processor recited in claim 10, wherein the at least one status bit is set high if the digital value is zero.

15. (Previously Presented) The system recited in claim 10, wherein the at least one status bit is set high if the digital value is a positive value.

16. (Previously Presented) The system recited in claim 10, wherein the at least one status bit is set high if the digital value is negative.

17. (Previously Presented) The system recited in claim 10, wherein the at least one status bit is set high if the digital value is a non zero value.

18. (Previously Presented) The system recited in claim 10, wherein the at least one status bit is set high based on the value of the least significant bit of the digital value.

19. (Previously Presented) In a pipelined instruction processor that executes instructions including conditional jump instructions, one or more of the

conditional jump instructions reading a digital value from memory to determine if the condition of the conditional jump instruction is satisfied, the improvement comprising:

a plurality of addressable registers, each of the addressable registers storing a value that includes a digital value and at least one jump status bit;

logic to access a current instruction, wherein the current instruction includes an address and a corresponding jump field, the address identifies one of the addressable registers and the corresponding jump field identifies a jump status bit of the at least one jump status bits within the identified addressable register;

a jump look-ahead controller for generating a jump look-ahead signal using the address that identifies one of the addressable registers and the jump field that identifies a jump status bit within the identified addressable register, the jump look-ahead signal is a function of the identified jump status bit;

tracking logic for tracking the addresses of a predetermined number of previous instructions in the pipelined instruction processor and comparing the addresses of each previous instruction to the address of the current instruction to generate a series of jump disable signals; and

conflict detection logic for generating a jump early signal using the jump look-ahead signal and the series of jump disable signals, the jump early signal initiates the conditional jump depending on the values of the jump disable signals.

20. (Previously Presented) The pipelined instruction processor as recited in claim 19, wherein each jump status bit is dependent on the digital value stored in the corresponding addressable register.

21. (Previously Presented) The pipelined instruction processor as recited in claim 19, further comprising a bit status generator for generating the corresponding jump status bits.

22. (Previously Presented) The pipelined instruction processor as recited in claim 19, further comprising a prediction logic block responsive to the jump early signal for implementing a prediction algorithm to predict the conditional jump depending on the values of the jump disable signals.

23. (Previously Presented) The pipelined instruction processor as recited in claim 19, wherein the tracking logic includes a queue for sequentially storing a pre-determined number of instructions prior to sequentially piping the pre-determined number of instructions through a read stage and decode stage in a pre-fetch pipeline.

24. (Previously Presented) The pipelined instruction processor as recited in claim 23, wherein the pre-determined number of instructions are sequentially

piped through an execution pipeline after being piped through a pre-fetch pipeline, the execution pipeline includes a write-back stage.

25. (Previously Presented) The pipelined instruction processor as recited in claim 24, wherein the addressable register is written during the write-back stage.

26. (Previously Presented) The pipelined instruction processor as recited in claim 25, wherein the execution pipeline further includes an address generation stage, a present address stage, an output operand stage, a capture data stage, and an arithmetic operation stage, all before the write-back stage.

27. (Previously Presented) A method for determine if a condition of a conditional jump instruction is satisfied in a pipelined instruction processor, the method comprising:

storing a digital value and one or more jump status bits that are based on the digital value in each of a plurality of address locations in an addressable memory;

accessing a current instruction, the current instruction having an address and a jump field, the address identifies a selected address location of the addressable memory, and the jump field identifies a selected jump status bit of the selected address location;

generating a jump look-ahead signal that is a function of the selected jump status bit read from the selected address location of the addressable memory, the identified jump status bit is accessed using the address and the jump field of the current instruction;

tracking the addresses of a predetermined number of previous instructions in the pipelined instruction processor and comparing the addresses to the address of the current instruction to generate a series of jump disable signals; and

generating a jump early signal using the jump-look ahead signal and the series jump disable signals, the jump early signal initiates a conditional jump depending on the value of the jump disable signals.